

金 沢 大 学 計 算 機 セ ン タ ー

利 用 の 手 引 き

S A S による統計分析（その 1）

ー プ ロ グ ラ ム の 入 力 ・ 実 行 か ら 出 力 ま で ー

執 筆 者 経 済 学 部 助 教 授

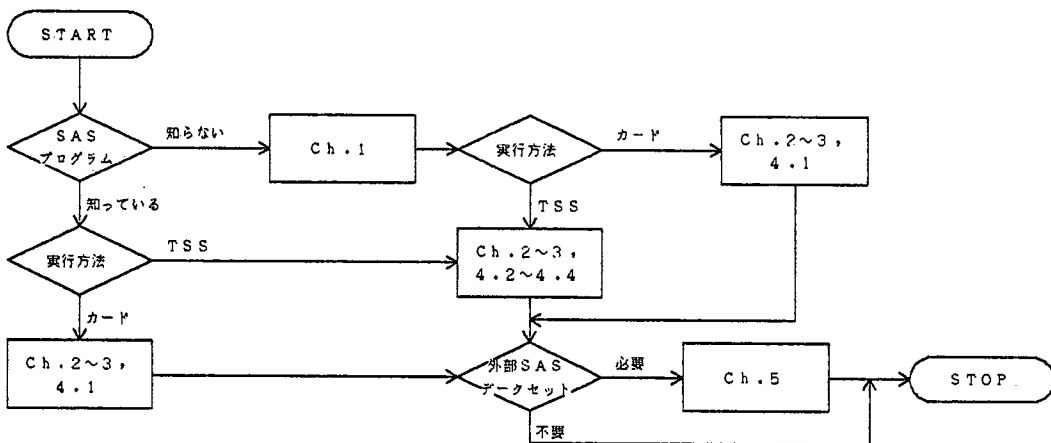
田 中 勝 人

はじめに

SAS (Statistical Analysis System) はその言葉が示す通り、本来は統計分析用のコンピュータ・システムである。しかし、SASは図形処理も簡単に行なえる利点があり、本学での利用目的は図形処理の傾向が強いようである。これはこれで結構であるが、統計分析だけを簡単に行なう必要性を感じている人々もかなりいるのではないかと思われる。

この手引きはそのような人々を対象にしてSASの使い方を第一歩で説明したものである。書くにあたっては、コンピュータを全く使ったことのない人にもわかるように心掛けたつもりである。全体は2つに分けられており、その1にあたる本手引きではSASプログラムの書き方から始めて、プログラムをコンピュータへ入力し、実行し、出力するまでの方法を説明してある。そして、その2ではSASを使って実際に統計分析する方法を個々の分析手法別に実例と共に示してある。

その2についてはもちろんであるが、本手引きでも最初から最後まで読み通す必要は全くない。参考のために本手引きの読み方のフローチャートを示しておく。図中で、Ch. 2～3, 4. 1などが四角で囲まれているのは、第2章、第3章、第4章1節を読むことを指示している。例えば、SASを全く知らなくてTSSで実行したい人は第1章から第3章、及び第4章2節から4節を読めばよい。そして、その2では分析目的にあった統計手法が説明されている節を読めば十分である。



本手引きでは図形処理の方法を一切説明していないが、それ以外についてもSASの機能がすべて網羅されているわけではない。従ってより深い知識を得たいと思われる人は以下の文献を参照されたい。

〔1〕 「SAS USER'S GUIDE」 SAS INSTITUTE INC. (1979) (日本語版あり)

〔2〕 「SAS VIEWS」 SAS INSTITUTE INC. (1980)

SASに関してはセンターからも次の2つの手引きが既に発行されている。

〔3〕 「SAS紹介—統計分析・図形処理システム—」 田中勝人著

〔4〕 「SAS (統計分析・図形処理) 使用方法」 車古正樹著

本手引きは、上の4つの文献、あるいは手引きを参考にして書かれている。

以下に挙げる手引きは、SASとは直接関係ないがTSSを初めて使う人のために有用である。

〔5〕 「バッチからTSSへ」 山崎光悦著

〔6〕 「FORTRANユーザのためのTSS入門」 中島恵美著

〔7〕 「TSS端末によるバッチ処理—フルスクリーン機能と出力検索—」 関崎正夫著

以上の手引きが本文中で引用される際は、左側の数字を使って例えば文献〔2〕、手引き〔6〕のように引用されているので注意されたい。

第1章 SAS概観

SASは簡単なステートメントを与えるだけで一切を実行してくれるコンピュータ・システムである。コンピュータ・システムとは、一緒に働くいくつかのコンピュータ・プログラムの集まりである。従って何らかの処理、分析を行なうためにわざわざ複雑なプログラムを書く必要はない。SASを使うためには、まず、そのような簡単なステートメントから成り立つSAS独自のプログラム（SASプログラム）の構造を理解する必要がある。

1. 1 SASプログラムの構成

SASプログラムは、他の言語によるプログラムに比べればはるかに単純な形をしている。プログラムの内容は、

- 1) DATAステップ：SASによる処理、分析用のデータセットを作るための一連のステートメント
- 2) PROCステップ：データセットを処理、分析するための一連のステートメント

の2つに大別される。これらのステップは、同一のSASプログラム内で何回でも使用できるし、どちらか一方のステップだけからなってもよい。SASプログラムには、これらのステップの他に、後の例（図1. 1）に示すようにデータを含むことができる。もちろん、データはSASプログラムの外部から入力することもできる。データがどのように入力されるかの記述も、DATAステップ内のステートメントで行なう。

DATA EXAMPLE;	#1
INPUT NAME\$1-15 SEX\$ SCORE;	#2
CARDS;	#3
TOM PHILLIPS M 65	#4
MARY JONES F 75	#4
DAVID KEYNES M 80	#4
JOHN STUART M 90	#4
PAMELA GILBERT F 85	#4
SUSAN YOUNG F 60	#4
PROC PRINT;	#5
PROC MEANS;	#6
DATA; SET EXAMPLE;	#7
IF SCORE<70 THEN DELETE;	#8
PROC SORT; BY SCORE;	#9
PROC PRINT;	#10

図1. 1 SASプログラムの例

DATA、及びPROCステップを構成する各ステートメントはFORTRANなどのプログラム言語よりもはるかに簡単なSAS言語で書かれ、必ずセミコロン（;）で終わる。ステートメントは一行（1カラム目から72カラム目）に複数個書いてもよい。従って、各ステートメントは何カラム

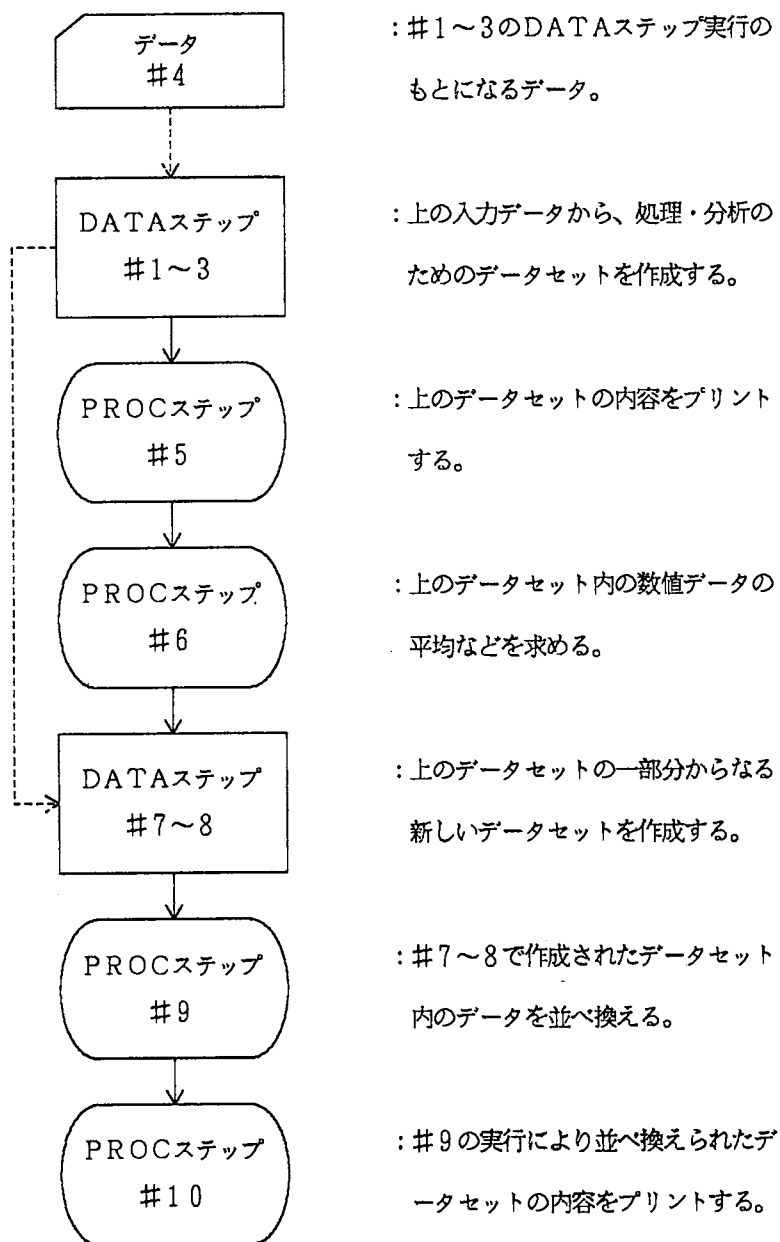


図1. 2 図1. 1のSASプログラムの流れと説明

目から書き始めてもよい。又、語句を途中で切るようなことがなければ一つのステートメントが複数行にわたってもよい。個々のステートメントの詳細については次章以降で説明することにして、図1. 1にプログラムの一例を示す。なお、右側の#1、#2、…は後の説明のため便宜上付けたもので、プログラムには関係ない。

図1. 1のSASプログラムの流れと簡単な説明を図1. 2に示す。

いかなるSASプログラムも、その形態は図1. 2と本質的に変わらず、単にステップ内のステートメントの数やステップの数が異なるだけである。

図1. 1のSASプログラムの内容は、図1. 2に簡単な説明を付けてあるのである程度理解できると思われる。しかし、各ステートメントについてはこの時点で理解できなくてもよい。このプログラムは次章以下で適宜引用されるのでそのたびに親しみのあるものになるであろう。そして、第3章まで読み進めば完全に理解できるであろう。

1. 2 SASデータセット

SASが処理、分析するデータは入力データそのものではなく、DATAステップにより作られたデータの集まりである。SASにより処理、分析可能となったデータの集まりをSASデータセットと呼ぶ。

SASデータセットは、データセット内にある変数名や属性を記述する部分と、データ部からなっている。データ部は、データを矩形の行列形式で配置する。その各行をオブザベーション、各列を変数と呼ぶ。オブザベーションには、最初から最後まで自動的に1, 2, ...とオブザベーション番号が付けられる。一方、変数にはDATAステップで指定した変数名が指定した順番で付けられる。変数名は、8文字以内で最初が英字かアンダラインで残りは英数字かアンダラインである。

例として、図1. 1の#1～3のDATAステップで作成されるSASデータセットのデータ部を表1. 1に示す。

表1. 1 SASデータセットの例

	NAME	SEX	SCORE
1	TOM PHILLIPS	M	65
2	MARY JONES	F	75
3	DAVID KEYNES	M	80
4	JOHN STUART	M	90
5	PAMELA GILBERT	F	85
6	SUSAN YOUNG	F	60

1. 3 データ入力について

SASは非常に広範囲の属性のデータを読むことができる。データの属性はDATAステップ内で

変数名と共に記述されるが、そのためのステートメントがINPUTステートメントである。通常の10進数及びE形式の数値データは変数名だけを指定すればよい。なお、数値データは倍精度でストアされる。文字データは、データ値が200字以内のものならば何でもよい。特に、8文字以内でブランクを含まない文字データは変数名だけを指定すればよい。

図1. 1の例では#2でINPUTステートメントが使われている。そこでは、3種類のデータが入力され、それぞれNAME、SEX、SCOREという変数名が付けられている。実際のデータは、#4で入力されているがNAMEとSEXは文字データなので変数名の直後に¥記号を付ける（¥記号のない機器では\$記号で代用できる）。又、SEXは1文字のデータ、SCOREは10進数データなので変数名以外に何も記述されていない。NAMEについては8文字を越えているので、1-15というように、入力カラムを指定するための記述がある。

その他の詳細は次章に譲るが、データの入力はINPUTステートメントの記述通りに行なわなければならないことを注意しておく。

1. 4 SASプログラムの実行について

図1. 1のようなSASプログラムを実行するためには、まずプログラムを入力しなければならない。詳細は第4章で説明するが、入力はカードでもTSS端末でもよい。

入力したSASプログラムは、カードの場合はJCL文と呼ばれるものを追加してカードリーダを通して、又、TSS端末の場合は一般にSAS用のコマンドを使って実行される。

実行結果は、カードの場合はLP（ラインプリンタ）、あるいはNLP（日本語ラインプリンタ）から出力され、TSS端末の場合は端末、端末用のプリンタ、LP、NLPなどに出力される。

以上の事はすべて第4章で説明されるが、参考のために図1. 1のSASプログラムをTSS端末で入力し、実行して結果を端末出力するための一つの方法を図1. 3に示しておく。図中、アンダラインを付けた部分が実際にキーボードから入力する部分である。この図の内容は、第4章2節で詳述するのでここでは触れないことにする。又、出力結果についてもここでは省略する。

```

READY
SP *
.....
1? DATA EXAMPLE;INPUT NAME$1-15 SEX$ SCORE;CARDS;
2? TOM PHILLIPS      M 65
3? MARY JONES        F 75
4? DAVID KEYNES      M 80
5? JOHN STUART       M 90
6? PAMELA GILBERT    F 85
7? SUSAN YOUNG       F 60
8? PROC PRINT;RUN;
   結果を端末に出力する
9? PROC MEANS;RUN;
   結果を端末に出力する
10? DATA;SET EXAMPLE;IF SCORE<70 THEN DELETE;
11? PROC SORT;BY SCORE;
12? PROC PRINT;RUN;
   結果を端末に出力する
13? /* (SASプログラムの実行を終了する)
.....
READY

```

図1. 3 SASプログラム実行のためのジョブストリームの例

第2章 DATAステップ

DATAステップは、SASデータセットを作成する一連のステートメントから成っている。SASデータセットを作成するためにはデータを入力しなければならないが、SASは多種多様なデータ入力に対応できるように、非常に多くのステートメントを備えている。

本章では8節にわたってこれらのステートメントを説明しているが、そのすべてを理解する必要はない。次節の表2. 1には、3つのデータ入力に応じて使われるステートメントが掲げられており、各自が必要とするステートメントを説明してある節を読めば十分である。

2. 1 DATAステップの構造

DATAステップは必ずDATAステートメントから始まる。それ以後の形式は、データがどこから入力されるかにより大きく3つに分けることができる。その3つとは次の通りである。

- i) データをSASプログラムと共に入力する。
- ii) データを外部OSデータセットから入力する。
- iii) データをSASデータセットから入力する。

ここで、外部OSデータセットとは、ユーザが登録できる区分あるいは順データセットであり、SASデータセットでないものを指す。各々の場合について、DATAステップで使われるステートメントを表2. 1に示す。

表2. 1 DATAステップで使われるステートメント

i) SASプログラム ステートメント	ii) 外部OSデータセット ステートメント	iii) SASデータセット ステートメント
DATA INPUT プログラミング 情報	DATA INFILE INPUT プログラミング	DATA SET プログラミング 情報

表2. 1 DATAステップで使われるステートメント

I) SASプログラム ステートメント	II) 外部OSデータセット ステートメント	III) SASデータセット ステートメント
CARDS	情報	

以下、それぞれのステートメントについて順を追って説明する。

2. 2 DATAステートメント

DATAステップの始まりをSASに知らせるものであり、ステップ実行後作成されるSASデータセット名を指定できる。さらに、SASデータセットを外部へ登録する場合はファイル名も指定する。

☆DATAステートメント

- ・DATAステップの始まりをSASに指示するステートメントである。
- ・ステートメントの形式には次の3通りがある。

DATA;

SASプログラム内だけで有効なSASデータセット（一時的SASデータセット）を作成する。作成されたデータセットにはSASが自動的に2語の名前を作成順にWORK.DATA1, WORK.DATA2, ...と付ける。ここで、WORKはファイル名、DATA1, DATA2, ...はSASデータセット名である。

DATA SASデータセット名;

上の場合と同様、一時的SASデータセットを作成する。SASデータセット名は英字かアンダラインで始まり、残りは英数字かアンダラインの8文字以内のものであればよい。ファイル名として自動的にWORKが指定される。

DATA ファイル名.SASデータセット名;

SASデータセットを外部へ登録する場合の形式である。このようにして作成されるSASデータセットを外部SASデータセットあるいは恒久SASデータセットと呼ぶ。ファイル名は英字で始まり残りは英数字の8文字以内のものであればよい。なお、SASデータセットの登録については第5章を参照されたい。

以後、上記3つのDATAステートメントにより付けられる名前を単にデータセット名と呼ぶ。第1章1節の図1. 1に示されたプログラムでは、#1と#7でDATAステートメントが使われ

ている。＃1では、EXAMPLEというSASデータセット名を指定しているが、＃7では何も指定していない。いずれにしろ、このようにして作成されるデータセットはSASプログラム終了時まで一時的に確保されるので、以後任意のステップで呼び出して処理できる。

2. 3 INPUTステートメント

データをSASプログラムと共に入力したり、外部OSデータセットから入力する場合には、入力するデータの変数名などをSASに対して記述する必要がある。そのために使われるのがINPUTステートメントである。

変数名は、第1章でも述べたように英字かアンダラインで始まり、残りは英数字かアンダラインの8文字以内のものである。

☆INPUTステートメント

- ・データをSASプログラムと共に入力する場合、あるいは外部OSデータセットから入力する場合に入力データの内容を記述するステートメントである。
- ・入力できるオブザベーションの数に制限はないが、変数の数は4000個以内である。
- ・文字変数の値は200字以内である。一方、数値変数はすべて実数とみなされる。10進数及びE形式以外の数値データを入力する時は、フォーマットを指定しなければならない。
- ・ステートメントの形式には次の3通りがある。

1) リスト入力

変数名だけを（文字変数には変数名の直後に\$記号を付けて）入力データ順に記述する形式である。

（例）

```
INPUT SEX$ AGE INCOME;
```

但しリスト入力が使えるのは次の条件をすべて満たす場合に限る。

- a) 各データ値の間にはブランクがある。
- b) 文字変数のデータにブランクがなく、8文字以内である。
- c) データに欠値がない。又は欠値がピリオドになっている。

2) カラム入力

データが占有するカラム位置を変数名の後に指定する入力形式である。

（例）

INPUT NAME¥1-20 SEX¥22 AGE 24-25;

3) フォーマット入力

データが占有するカラム数と小数部の桁数を指定し、さらに必要な場合にはデータのフォーマット名も指定する形式である。数値変数に対しては、変数名の後1カラムあけて

w . d : 10進数、E形式のデータ

フォーマット名 w . d : 上記以外のデータ

のように書く。但し、wは入力データの占有カラム数（あるいはバイト数）、dは小数部の桁数である。なお、d=0の時は省略できる。フォーマット名にはHEX（2進固定小数点）、IB（2進浮動小数点）、RB（浮動小数点実数）や、日付けなどの特殊なフォーマットがある（後者については本章8節を参照されたい）。文字変数に対しては、変数名の後に次のように入力する形式がある。

¥w. : 通常の文字データ

¥CHARw. : 値の前後のブランクを有効とする

¥HEXw. : 16進文字データ

¥VARYINGw. : 可変長文字データ

さらに、変数名の前にデータが占有するカラムの先頭カラムを@記号と共に指定する形式や、先頭カラムの相対位置を+記号と共に指定する形式がある。

(例)

INPUT NAME¥20. +1 SEX¥1. @24 AGE 2. INCOME 5.;

この例では、最初の20カラムにNAMEの値、1カラム右へ行って22カラム目にSEXの値、24カラムから25カラム目にAGEの値、次の5カラム内（26～30カラム）にINCOMEの値が入力される。

- ・ステートメントの中で3つの入力形式を同時に使用してもよい。

(例)

INPUT NAME¥1-20 SEX¥ +1 AGE INCOME 5.;

- ・1個のオブザベーションを複数行にわたって入力する場合には、/、あるいは#記号を用いる。

INPUTステートメントで使われる@, +, /, #はポインター機能と呼ばれる。

(例)

INPUT NAME¥1-20 SEX¥ AGE/INCOME;

INPUT NAME¥1-20 #2 SEX¥ AGE

#3 INCOME;

最初の例は、オブザベーションを2行にわたって入力するものである。後者の例は3行にわたって入力するもので、2行目以降は#記号の直後に行番号を付ける。

- ・INPUTステートメントの特殊な入力形式を以下に示す。

```
INPUT X Y @@;
```

X, Yのデータを同一行に交互にリスト入力する。例えば、Xのデータが1, 2, ..., Yのデータが10, 20, ...の時、1 10 2 20 ...のように入力する。

```
INPUT X1-X10;
```

変数X1, X2, ..., X10のデータをリスト入力する。

```
INPUT(X1-X10)(2.);
```

変数X1, X2, ..., X10のデータを2カラムずつ入力する。

```
INPUT(NAME X1-X10)(¥10. 10*4.);
```

変数NAMEの値を1~10カラムに入力し、次にX1~X10の10個の値を4カラムずつ入力する。

第1章1節の図1. 1では#2でINPUTステートメントが使われている。ここでは3つの変数NAME, SEX, SCOREが記述されているが、文字変数NAMEはカラム入力でデータは15カラムまでに入力される。文字変数SEXと数値変数SCOREはリスト入力で、16カラム目（以降）にSEXのデータを入力し、1カラム（以上）あけてSCOREのデータを入力することになる。

2. 4 CARDSステートメント

データをSASプログラムと共に入力する場合には、INPUTステートメントの他にCARDSステートメントが必要である。CARDSステートメントはDATAステップの最後に与え、直後にデータを入力する。

☆CARDSステートメント

- ・データをSASプログラムと共に入力することをSASに知らせるためのステートメントである。

- ・一般形は

```
CARDS;
```

であり、直後にデータを入力する。

- ・データ入力後の最初のステートメントが長すぎてセミコロンを含むことができない時は、そのステートメントの前にセミコロンを挿入する。

以上3つの節でDATA, INPUT, CARDSステートメントについて説明したが、この3つ

のステートメントはデータをSASプログラムと共に入力する場合に必要な不可欠のものである。再び第1章1節の図1. 1に戻って#1～4の内容を理解されたい。

2. 5 INFILEステートメント

データを外部OSデータセットから入力する場合には、SASに対してそれを知らせるためのINFILEステートメントが必要である。ここで、外部OSデータセットとは、前述したように、ユーザが登録できる区分あるいは順データセットであり、SASデータセットでないものを意味する。

INFILEステートメントはINPUTステートメントの前に置かなければならない。

☆INFILEステートメント

- ・SASデータセットでない外部OSデータセットからデータを入力することをSASに知らせるためのステートメントである。

- ・一般形は

INFILE ファイル名 オプションの並び;

である。ファイル名は、入力データセットのファイル名であり英字で始まり残りは英数字の8文字以内のものである。

- ・オプションには次のものがある。

LS=カラム数

各レコードのうち、読み込ませたいカラム数を指定する。例えば、各レコードの73～80カラムにシーケンス番号がある場合にはLS=72とする。ここで、レコードとは入力データの各行を意味する。

FIRSTOBS=レコード番号

指定されたレコード番号からのデータを処理、分析の対象とする。下のOBSオプションを指定しないと最後のレコードまでのデータを対象とする。

OBS=レコード番号

指定されたレコード番号までのデータを処理、分析の対象とする。上のFIRSTOBSオプションを指定しないと最初のレコードからのデータを対象とする。

- ・SASプログラムの中に、入力データセット名を書く必要はない。それは実行の際、バッチジョブの場合にはJCL文で指定し、TSSではコマンドで指定する。
- ・ステートメントの例を下に示す。

INFILE FF LS=72 FIRSTOBS=11 OBS=90;

ここでFFがファイル名である。各レコードは72カラムまでを読み込み、11番目のレコー

ドから始めて90番目のレコードまでのデータを処理、分析の対象とすることになる。

2. 6 SETステートメント

データをSASデータセットから入力する場合はSETステートメントを使う。SASデータセットには、同一のSASプログラム内で作成された一時的データセットと、外部に登録された恒久データセットの2つがあり、ステートメントの形式もそれぞれ異なっている。

☆SETステートメント

- ・データをSASデータセットから入力することをSASに知らせるためのステートメントである。

- ・一般形は

SET SASデータセット名;

あるいは

SET ファイル名.SASデータセット名;

である。前者は一時的SASデータセットから入力する場合、後者は恒久SASデータセットから入力する場合である。

- ・データセット名の後に、処理、分析の対象とするオブザベーション番号の最初と最後をFIRSTOBS、OBSオプションで指定できる。

(例)

SET SURVEY(FIRSTOBS=11 OBS=200);

これにより一時的SASデータセットSURVEYのオブザベーションのうち、11番目から200番目までが処理、分析の対象となる。FIRSTOBSオプションを省略すれば最初から、又、OBSオプションを省略すれば最後までが対象となる。

- ・入力するSASデータセットが同一の変数からなっていれば、複数個のデータセット名を指定できる。実行後、すべてのオブザベーションを含む1つのSASデータセットが作成される。
- ・SASデータセットを入力してもオブザベーションを全く変更しない場合はDATAステップは不要であり、即座にPROCステップを実行すればよい。

(例)

DATA;SET EXAMPLE;PROC PRINT;は、

PROC PRINT DATA=EXAMPLE;としてよい。

第1章1節の図1. 1に示されたSASプログラムでは、#7でSETステートメントが使われている。ここでは一時的SASデータセットEXAMPLEが入力されているが、その目的はオブザベーションに変更を加えて処理、分析することにある。#8がオブザベーションに変更を加えるためのステートメントであり、次節で説明する。

以上で3つのデータ入力方法に応じて使われるステートメントの説明を終える。以下、2つの節では入力データを加工したり新たに変数を追加したりするためのステートメントや、出力の体裁を整えるためのステートメントをとり上げる。

2. 7 プログラミング・ステートメントとSAS関数

分析目的によっては入力データの一部だけを分析したり、あるいは新たに変数を追加して分析したい場合もある。本節では、まずこのような場合に使われるステートメントのいくつかを説明する。これらは一括してプログラミング・ステートメントと呼ばれる。

本節後半ではプログラミング・ステートメントの中で使うことのできるSAS関数について述べる。

☆プログラミング・ステートメント

- ・データセットの一部だけを分析したり、変数を追加したりする場合に使われるステートメントである。
- ・ステートメントには以下のものがある。

(1) 代入ステートメント

新しく変数を作ったり、既に存在している変数の値を変更したりする。一般形は
変数名=数式（あるいはデータ値）；

である。

(例)

```
X=A+B-C*D/E;
```

```
Y=NAME;
```

これらの例では、右辺の変数は既存でなければならない。左辺の変数は既存でなくてもよい。前者の例は数値変数に限るが、後者は数値変数でも文字変数でもよい。

(例)

```
V=1;
```

```
W='GOOD';
```

これらの例は共に、データ値そのものを代入する例である。前者の例は数値を代入するものであり、後者は文字データを代入する場合である。

(2) SUMステートメント

自動的に初期化を行ない累積和を求める。一般形は
変数名+数式(あるいは数値);

である。

(例)

`X+A*A;`

既存の変数Aの2乗の累積和を変数Xに入れる。Xは既存でなくてもよい。

(例)

`NX+B*=.;`

`LW+NAME*= ' ' ;`

論理式を含む例であり、前者は数値変数Bの値が欠値でない場合にNXに1を入れ、欠値の場合0を入れて累積和を求める。後者も同様であるが、文字変数の例であり、文字変数の欠値はこのように2つの引用符で表す。なお、2つの引用符の間にはブランクを入れなくてもよい。

(3) IFステートメント

オブザベーションの一部だけを分析する。一般形は

IF 式の表現;

あるいは

IF 式の表現 THEN ステートメント(あるいは式の表現);

である。後者では、さらに

ELSE ステートメント(あるいは式の表現);

が追加される場合が多い。

(例)

`IF SCORE<70 THEN DELETE;`

これは第1章1節の図1. 1に示したプログラムの#8である。SCOREの値が70より小さいものをデータセットから除外するためのステートメントである。これは

`IF SCORE>=70;`

と同じである。

`IF X>=80 THEN CLASS='A';`

`IF 60<=X<80 THEN CLASS='B';`

`ELSE CLASS='C';`

Xの値が80以上の時CLASSの値をA、60以上80未満の時Bとし、それ以外の時Cとする。

(4) OUTPUTステートメント

入力データの各レコードから複数個のオブザベーションを作ったり、複数個のオブザベーションから1個のオブザベーションを作るためのステートメントである。一般形は

OUTPUT;

である。

通常のDATAステップでは、DATAステートメントから最後のステートメントまでを反復して、1個ずつオブザベーションを作って行く。この場合、反復回数は作成されるオブザベーションの個数に等しい。しかし、OUTPUTステートメントを使うことにより、多様なデータセットが作成される。

(例)

```
DATA; INPUT NAME$ X1-X3;
      X=X1; OUTPUT; X=X2; OUTPUT; X=X3; OUTPUT;
CARDS;
A   1   2   3
B  10  20  30
```

このDATAステップにより作成されるSASデータセットは次のようになる。

NAME	X1	X2	X3	X
A	1	2	3	1
A	1	2	3	2
A	1	2	3	3
B	10	20	30	10
B	10	20	30	20
B	10	20	30	30

上の例で、OUTPUTステートメントを全く使わなければ2個のオブザベーションを持つデータセットが作成される。そして、変数Xの値はX3と同じになる。

OUTPUTステートメントの有用な例を、さらに次の(5)で示す。

(5) DOステートメント

対応するENDまでのステートメントを反復実行する。一般形は

DO 変数名=初期値 TO 最終値 BY 増分;

である。増分(負の増分でもよい)が1の時はBY以下は不要である。他のプログラミング・ステートメントなどを入力後

OUTPUT;END;

とする。

(例)

```
DATA;DO X=0 TO 10 BY .05;
```

```
Y=X*X;OUTPUT;END;
```

0から10まで0.05きざみで2乗の値を求めデータセットに入れる。この場合、OUTPUTステートメントがあるので、X、Yすべての値がデータセットに入る。OUTPUTステートメントがない場合は、1個のオブザベーションからなるデータセットが作成される。

(例)

```
DATA;DO YEAR=1801 TO 1980;INFILE FF LS=72;
```

```
INPUT X Y @@;OUTPUT;END;
```

これにより作成されるデータセットは、3つの変数YEAR、X、Yからなり、180個のオブザベーションを含む。外部OSデータセットから入力されるデータはXとYだけであるが、DO、OUTPUTステートメントを使って変数YEARのデータが作られている。

(6) RETAINステートメント

直前のオブザベーションの値を保持する。一般形は

```
RETAIN 変数名 初期値;
```

である。変数名、初期値は複数個指定してもよい。指定された変数の値だけが保持される。

(例)

```
DATA;RETAIN S 0;
```

```
DO I=1 TO 100;S=S+I**4;OUTPUT;END;
```

これにより作成されるデータセットは、変数SとIからなり、100個のオブザベーションからなる。変数Sには1から100までの4乗の累積和が入り、Iには1から100までの値が入る。RETAINステートメントがないと、Sの値はすべて欠値となる。

(7) DROPステートメント

データセットから特定の変数を取り除く。一般形は

```
DROP 変数名の並び;
```

である。なお、同一のDATAステップで(8)のKEEPステートメントと一緒に使ってはならない。

(例)

```
DATA;INPUT A B C D;
```

```
X=A+B;DROP A B;
```

(8) KEEPステートメント

特定の変数だけをデータセットに入れる。一般形は

```
KEEP 変数名の並び;
```

である。同一のDATAステップで(7)のDROPステートメントと一緒に使ってはならない。

(例)

```
DATA; INPUT A B C D;  
X=A+B-C/D; KEEP X;
```

本節で説明したプログラミング・ステートメントを使ってSASデータセットを作成する簡単な例を以下に示す。

(a)

```
DATA CLASS; INPUT NAME$ SEX$  
AGE SCORE1 SCORE2; CARDS;
```

データ

上のDATAステップで作成されたSASデータセットCLASSが表2. 2に示されている。

表2. 2 SASデータセットCLASS

	NAME	SEX	AGE	SCORE1	SCORE2
1	M1	M	19	60	100
2	F1	F	20	70	90
3	F2	F	18	80	80
4	M2	M	21	90	70
5	F3	F	22	100	60

(b)

```
DATA B; SET CLASS;  
SCORE=SCORE1+SCORE2;  
T_SCORE1+SCORE1; T_SCORE2+SCORE2;  
DROP SCORE1 SCORE2;
```

SASデータセットCLASSの変数SCORE1とSCORE2の和を変数SCOREに入れ、SCORE1とSCORE2の累積和をそれぞれT_SCORE1, T_SCORE2に入れる。そして、SCORE1とSCORE2はSASデータセットBから除外する。

以上により作成されるデータセットBは表2. 3のようになる。

表2. 3 SASデータセットB

	NAME	SEX	AGE	SCORE	T_SCORE1	T_SCORE2
1	M1	M	19	160	60	100

表2. 3 SASデータセットB

	NAME	SEX	AGE	SCORE	T_SCORE1	T_SCORE2
2	F1	F	20	160	130	190
3	F2	F	18	160	210	270
4	M2	M	21	160	300	340
5	F3	F	22	160	400	400

(c)

DATA C;SET CLASS;

IF SEX='M' THEN DELETE;

BIRTHYR=1983-AGE;DROP AGE;

SASデータセットCLASSの変数SEXの値がMのものを除外し、新たにBIRTHYRを上のように定義して追加する。そして変数AGEを除外する。このようにして作成されたデータセットCは表2. 4のようになる。

表2. 4 SASデータセットC

	NAME	SEX	SCORE1	SCORE2	BIRTHYR
1	F1	F	70	90	1963
2	F2	F	80	80	1965
3	F3	F	100	60	1961

次に、プログラミング・ステートメントの中で使うことのできるSAS関数について説明する。SAS関数とは、四則演算と同様にSASプログラムの中で自由に使える関数である。主要なSAS関数の例を表2. 5に示す。なお、表の中で引数は特に断らない限り数値でも変数名でもよい。

表2. 5 主要なSAS関数一覧

関数の形式	内 容
LAGn (引数)	ラグnの値を求める。nは100まででn=1の時は省略してよい。
DIFn (X)	$DIFn(X) = X - LAGn(X)$ を求める。
ABS (引数)	絶対値を求める。
CEIL (引数)	引数より小さくない最小整数を求める。
FLOOR (引数)	引数を越えない最大整数を求める。

表 2. 5 主要なSAS関数一覧

関数の形式	内 容
INT (引数)	小数部分を切り捨てる。引数が正の時FLOOR、負の時CEILと同じである。
MAX (引数の並び)	引数のうちの最大値を求める。
MIN (引数の並び)	引数のうちの最小値を求める。
MOD (引数1, 引数2)	引数1を引数2で割った時の剰余を求める。
SQRT (引数)	正の平方根を求める。
COS (引数)	余弦を求める。逆余弦はARCOS (引数)、双曲余弦はCOSH (引数) である。
SIN (引数)	正弦を求める。逆正弦はARSIN (引数)、双曲正弦はSINH (引数) である。
TAN (引数)	正接を求める。逆正接はATAN (引数)、双曲正接はTANH (引数) である。
EXP (引数)	eの指数関数の値を求める。
ERF (引数)	$2 \exp(-X^2) / \sqrt{\pi}$ の0から引数の値までの積分値を求める。
GAMMA (引数)	ガンマ関数の値を求める。
LOG (引数)	自然対数の値を求める。
LOG10 (引数)	常用対数の値を求める。
PROBIT (引数)	標準正規分布の分位点を求める。
PROBNORM (引数)	標準正規分布の下側確率を求める。
PROBCHI (X, DF)	自由度DFの中心カイ2乗分布に従う確率変数がX以下となる確率を求める。
PROBT (X, DF)	自由度DFの中心t分布に従う確率変数がX以下となる確率を求める。
PROBF (X, M, N)	自由度 (M, N) の中心F分布に従う確率変数がX以下となる確率を求める。
PROBGAM (X, A)	パラメータAのガンマ分布に従う確率変数がX以下となる確率を求める。
GAMINV (P, A)	パラメータAのガンマ分布の100P%分位点を求める。
PROBBETA (X, A, B)	シェイプパラメータA, Bのベータ分布に従う確率変数がX以下となる確率を求める。

表2. 5 主要なSAS関数一覧

関数の形式	内 容
POISSON (L, N)	平均Lのポアソン分布に従う確率変数がN以下となる確率を求める。
PROBBNML (P, N, M)	パラメータP (成功確率) N (試行回数) の2項分布に従う確率変数がM以下となる確率を求める。
TPROB (X, DF, C)	自由度DF, 非心度Cのt分布に従う確率変数がX以下となる確率を求める。
TINV (P, DF, C)	自由度DF, 非心度Cのt分布に従う確率変数の100P%分位点を求める。
TNONCT (X, DF, P)	自由度DFの非心t分布の100P%分位点がXとなるような非心度を求める。
CPROB (X, DF, C)	自由度DF, 非心度Cのカイ2乗分布に従う確率変数がX以下となる確率を求める。
CINV (P, DF, C)	自由度DF, 非心度Cのカイ2乗分布に従う確率変数の100P%分位点を求める。
CNONCT (X, DF, P)	自由度DFの非心カイ2乗分布の100P%分位点がXとなるような非心度を求める。
FPROB (X, M, N, C)	自由度 (M, N)、非心度CのF分布に従う確率変数がX以下となる確率を求める。
FINV (P, M, N, C)	自由度 (M, N)、非心度CのF分布に従う確率変数の100P%分位点を求める。
FNONCT (X, M, N, P)	自由度 (M, N) の非心F分布の100P%分位点がXとなるような非心度を求める。
UNIFORM (引数)	0と1の間の一様乱数を発生する。引数は0, 5, 6のいずれかの値、又は7桁の奇整数である。
NORMAL (引数)	標準正規乱数を発生する。引数は0, 5, 6のいずれかの値、又は7桁の奇整数である。

SAS関数を使ったDATAステップの例を以下に示す。次のSASプログラムを考えよう。

```
DATA FUNCTION; INPUT X Y Z; LX=LAG(X);
  MAXIMUM=MAX(X,Y,Z); CARDS;
10 60 40
50 20 90
```

80 70 30

DATAステップ実行後、SASデータセットFUNCTIONは次のようになる。

	X	Y	Z	LX	MAXIMUM
1	10	60	40		60
2	50	20	90	10	90
3	80	70	30	50	80

2. 8 情報ステートメントと特殊なフォーマット

本節ではSASデータセットの作成には直接関係ないが出力の体裁を整えるために有用なステートメントであるLABEL, FORMATについて述べる。

☆LABELステートメント

- ・変数にラベルを与えるためのステートメントである。
- ・一般形は
LABEL 変数名=ラベル;
である。ラベルはブランクを含めて40字以内である。
- ・1つのステートメントで複数個の変数にラベルを定義できる。
- ・ラベルが等号やセミコロンを含む場合はラベル全体を引用符で囲み、ラベルが1つの引用符を含む場合は2つの引用符を書く。
- ・以下に例を示す。

```
LABEL SCORE=GRADE ON FIRST YEAR TEST  
R=MOTHER'S RESPONSE  
GNP='GNP=GROSS NATIONAL PRODUCT';
```

☆FORMATステートメント

- ・変数にフォーマットを指定するためのステートメントである。
- ・一般形は
FORMAT 変数名1 フォーマット 変数名2 フォーマット・・・;
である。

- ・複数の変数に同一のフォーマットを指定する場合は変数名を並べた後に1回だけフォーマットを指定すればよい。
- ・第2章3節のINPUTステートメントの所で述べた以外のフォーマットには次のものがある。なお、wは占有カラム数、dは小数部の桁数を示す。

COMMAw, d : 数値に3桁おきにコンマを付ける。
 ROMANw. : ローマ数字に変換する。
 WORDSw. : 数字を言葉(英語)に変換する。
 DATEw. : 日付けを04 JUL 1983のようにする。
 YYMMDDw. : 日付けを83-07-04のようにする。
 MMDDYYw. : 日付けを7/4/83のようにする。
 MONYYw. : 日付けをJUL 83のようにする。
 WEEKDATEw. : 日付けをMONDAY, JULY 4, 1983のようにする。
 WORDDATEw. : 日付けをJULY 4, 1983のようにする。

- ・ステートメントはPROCステップ内で使ってもよい。例えば、SASデータセットの内容をプリントするPRINTプロシージャの中で使うと、変数値が指定されたフォーマットで出力される。

LABEL, FORMATステートメントの実例の使用例については、第4章1節、第7章10節などで示す。

2.9 変数__N__について

SASデータセットが作成される際には、オブザベーション番号のデータが変数__N__として自動的に生成される。しかし、__N__はDATAステップ実行後には消去される。従って、DATAステップ内においてのみプログラミング・ステートメントなどで使うことができる。以下に例を示す。

```
N=__N__; オブザベーション番号をデータ値として変数Nに入れる。
IF __N__ <= 10; オブザベーション番号が10以下のものだけをデータセットに入れる。
IF MOD(__N__, 3) = 1
THEN DELETE; オブザベーション番号が1, 4, 7, ... (3で割って余りが1となるもの)
をデータセットから除外する。
```

第3章 PROCステップ概観

PROCステップはDATAステップで作成されたSASデータセットを処理、分析する一連のステートメントから構成されている。

DATAステップがDATAステートメントから始まるように、PROCステップはPROCステートメントから始まる。

PROCステートメントの一般形は

PROC プロシージャ名 オプションの並び;

である。ここで、プロシージャ名とは個々の統計分析の手法に対してSASが付けた名前であり必ず指定されなければならない。一方、オプションは必要な場合にのみ指定すればよい。

PROCステップは上述のPROCステートメントだけで構成される場合も多いが、プロシージャによっては付帯情報を持ったステートメントを必要とする場合も少なくない。さらに、多様な分析をするためにいくつかの追加ステートメントを与えることができる。

3.1 SASプロシージャ一覧

SASデータセットを処理、分析するためのSASプロシージャは非常に多岐にわたっている。まず、主要な統計分析用のプロシージャを目的別に表3.1に示す。表の中でプロシージャ名の後に○印を付けたものは第7章以降で使用方法が説明されているプロシージャである。

表3.1 主要な統計分析用SASプロシージャ

プロシージャ名		実 行 内 容
(A)		データの要約
CHART	○	ヒストグラム、円グラフ、スターチャートの作成
CORR	○	各種の相関係数、及び有意確率の計算
FREQ	○	分割表の作成、及び独立性の検定
FORMAT	○	データ値のフォーマット指定

表3. 1 主要な統計分析用SASプロシージャ

プロシージャ名		実 行 内 容
MEANS	○	平均、標準偏差、最大、最小値などの計算
PLOT	○	2個の変数のプロット、及び3個目の変数の等高プロット
RANK	○	データ値の順位づけ
STANDARD	○	数値データの標準化
TTEST	○	2標本におけるt検定
UNIVARIATE	○	モード、分位点、頻度などの計算
(B)		回帰分析
LAV		最小絶対偏差法による回帰分析
LOGIST	○	ロジットモデルの推定
NLIN	○	非線形回帰モデルの推定
PROBIT	○	プロビットモデルの推定
REG	○	一般線形回帰モデルの分析
RIDGREG		リッジ法による回帰分析
RSQUARE	○	説明変数のすべての組合せによる回帰分析
STEPWISE	○	最適な説明変数選択を伴う回帰分析
(C)		分散分析
ANOVA	○	つりあい型データの分散分析
GLM	○	一般の線形モデルの分散分析
NESTED		変動効果がネストされている場合の分散分析
NPAR1WAY		順位スコアの分散分析
VARCOMP		変量模型の分散分析
(D)		時系列分析
ARIMA	○	ボックス・ジェンキンス型時系列モデルの推定、予測
AUTOREG	○	誤差項が自己回帰過程に従うモデルの推定
FORECAST		時系列の予測
SPECTRA	○	スペクトル分析
STATESPACE		状態空間モデルの推定、予測
(E)		多変量解析
CANCORR		正準相関分析
CLUSTER	○	クラスター分析
DISCRIM		判別分析

表3. 1 主要な統計分析用SASプロシージャ

プロシージャ名		実 行 内 容
FACTOR	○	因子分析
FUNCAT		多項応答関数の線形化と推定
GUTTMAN		ガットマン尺度モデルの分析
NEIGHBOR		最近隣法による判別分析
PLAN		無作為実験計画の作成
PRINCOMP	○	主成分分析
(F)		計量経済分析
MODEL		非線形モデルの特定化
SIMLIN		線形連立方程式モデルによるシミュレーション
SIMNLIN		非線形連立方程式モデルによるシミュレーション
SYSNLIN		非線形連立方程式モデルの推定
SYSREG	○	線形連立方程式モデルの推定
X11		X-11法による季節変動の調整

次に表3. 2にSASデータセット管理のための主要なプロシージャを示す。プロシージャ名の後に○印を付けたものは第5章、あるいは第6章で使用方法が説明されているプロシージャである。

表3. 2 SASデータセット管理のための主要なプロシージャ

プロシージャ名		実 行 内 容
CONTENTS	○	SASデータセットの物理的特性や内容のプリント
CONVERT		他のシステムファイル（例えばSPSS）をSASデータに変換
COPY	○	SASデータセットのコピー
DATASETS	○	外部SASデータセットのリスト、削除、改名
DELETE	○	SASデータセットの削除
EDITOR	○	SASデータセットの編集
PRINT	○	SASデータセット内のデータを変数名と共にプリント
SORT	○	SASデータセット内のオブザベーションを再配列

3. 2 PROCステップ実行上の特徴

前節で述べたように、PROCステップの最も簡単な形式は

PROC プロシージャ名;

である。この場合の実行上の特徴は次の通りである。

- (1) 最後に作成されたSASデータセットを処理する。
- (2) オブザベーション全部を処理する。
- (3) すべての変数（計算プロシージャに対しては数値変数のみ）を処理する。

従って(1)～(3)以外の方法で実行したい場合は、そのためのオプション、あるいは追加的なステートメントが必要となる。それを以下に例と共に述べる。

(a) 特定のデータセットを処理する場合

(例) PROC PRINT DATA=データセット名;

指定されたデータセット内のデータを変数名と共にプリントする。ここでデータセット名は、一時的SASデータセットの場合は1語の名前、恒久SASデータセットの場合は2語の名前となる（第2章2節のDATAステートメントを参照）。

(b) オブザベーションの一部を処理する場合

一般的にはDATAステップでプログラミング・ステートメントを使えばよいが、次のような場合にはPROCステップでもできる。

(例) PROC MEANS DATA=データセット名(FIRSTOBS=m OBS=n);

指定されたデータセットのオブザベーションのうち、m番目からn番目のオブザベーションを取り出し、その中の数値変数に対して平均、標準偏差などを求める（FIRSTOBS, OBSオプションについては、第2章6節のSETステートメントを参照）。

(c) 特定の変数を処理する場合

(例) PROC MEANS;VAR HEIGHT WEIGHT;

直前に作成されたSASデータセットの変数のうち、HEIGHT, WEIGHTだけにMEANSプロシージャを実行する。

なお、上の(a)～(c)を同時に使うことも可能である。

(例) PROC MEANS DATA=SURVEY(FIRSTOBS=11 OBS=90);
VAR HEIGHT WEIGHT;

3. 3 追加ステートメントを必要とするプロシージャについて

プロシージャの中にはPROCステートメント以外にも、付帯情報を持ったステートメントを必ず

伴わなければならないものが少なくない。例えば、SASデータセット内のオブザベーションを再配列するためのSORTプロシージャは必ずBYステートメントを伴わなければならない。第1章1節の図1. 1に示されたプログラムにも#9で

```
PROC SORT;BY SCORE;
```

のように使われている。この場合、オブザベーションの再配列は数値変数SCOREがとる値の小さい方から行なわれる。BYステートメントで指定される変数は文字変数でもよい。その場合にはSASで決められた大小順で、やはり小さい方から配列される。又、指定する変数は複数個でもよい。その時は、左側の変数から再配列を行なう（詳細は第6章3節のSORTプロシージャを参照されたい）。

別の例として、データをプロットするためのPLOTプロシージャではプロットする変数を指定するためのPLOTステートメントが必要である。例えば

```
PROC PLOT DATA=A;
```

```
  PLOT Y*X;
```

とすると、Yの値を縦軸にXの値を横軸に持つような座標軸が自動的にスケーリングされ、(Y, X)の値がプロットされる（詳細は第7章2節のPLOTプロシージャを参照されたい）。

本章はPROCステップを概観しただけであるが、以上の説明で第1章1節の図1. 1に示したSASプログラムは完全に理解できると思われる。

第4章 SASプログラムの入力、実行、出力

SASプログラムの入力、実行、及び出力はバッチでもTSSでも可能である。どちらで行なうかはユーザの選好にもよるが、TSSの方が便利である。しかし、実行時間が長くなるジョブや、大量の出力を要するものはバッチで行なう方がよい。なお、ここではTSS端末の使い方については述べないので初心者は手引き〔5〕,〔6〕,〔7〕を参照されたい。

本章ではデータをSASプログラムと共に入力する場合と外部OSデータセットから入力する場合についてバッチ及びTSSによる入力、実行、出力方法を説明する。恒久SASデータセットの場合については次章で述べる。

4.1 バッチによる入力と実行

パンチカードあるいはTSSのSUBMITジョブによるSASプログラム実行のためのジョブストリームを図4.1に示す。なお、図中のSASプログラムはデータをSASプログラムと共に入力してPLOTプロシージャを実行するものである。

☆図4.1の説明

#1, 2, 4:データをSASプログラムと共に入力する場合に必要となるJCL文。
AB9999Aはジョブ名と呼ばれ、最初の6文字はユーザ識別修飾子、最後のAはジョブ識別子である。以後、AB999をユーザ識別修飾子として使う。

//AB9999A JOB ,PASS=XXXXXXXX, CLASS=X	#1
// EXEC SAS	#2
//SAS.SYSIN DD *	#4
TITLE RESPONSE VS. TIME;	#5
TITLE2 SIMULATED DATA;	#5
DATA;INPUT T Y;	#6
LABEL T=TIME Y=RESPONSE;	#7
CARDS;	#8
1 15	#9
2 26	#9
3 34	#9
4 47	#9
5 55	#9
PROC PLOT;PLOT Y*T;	#10
/*	#11
//	#12

図4.1 バッチのためのジョブストリーム

#5:出力ページに標題を付けるためのTITLEステートメント(詳細は第6章3節を参照)。

＃6～9：DATAステップ及びデータ。＃7のLABELステートメントで指定されるラベルが＃10で実行されるPLOTプロシージャにより作成される座標軸の軸名となる。

＃10：PLOTプロシージャの実行。

＃11, 12：SASプログラムの後に必要となるJCL文。

データを外部OSデータセットから入力する場合には、図4. 1の＃2と＃4の間にDD文と呼ばれるJCL文を追加する必要がある。例えば、データを区分データセットAB9999.KANAZAWA.DATA (NO1) から入力する場合には次の＃3を追加する。

```
//STUDY DD DSN=AB9999.KANAZAWA.DATA(NO1),DISP=SHR      #3
```

ここでSTUDYはファイル名である。これに伴い、SASプログラムのDATAステップにINFILISTステートメントが必要となるが、STUDYはそのファイル名と同一でなければならない。

ジョブの実行は、パンチカードの場合にはカードリーダーで、又TSSによるバッチ処理ではSUBコマンドあるいはSUBサブコマンドを使う。

カードリーダーによるジョブの出力は、NLPあるいはLPへ、又SUBMITジョブによる場合はこれらの他にプリンタや端末に出力できる。なお、SUBMITジョブの詳細については手引き〔7〕を参照されたい。

4. 2 TSSによる入力と実行（端末出力の場合）

本学計算機センターで作成されたSPコマンドを使って結果を端末に出力する方法を説明する。端末以外へ出力する場合にもSPコマンドで同様にできるが、これについては次節で説明する。

まず、データをSASプログラムと共に入力する場合について述べる。

4. 2. 1 データをSASプログラムと共に入力する場合

表4. 1 結果を端末出力するためのSPコマンド
(データをSASプログラムと共に入力する場合)

	入 力 方 法	SASプログラム
(1)	SP *	一時的
(2)	SP EXAMPLE.DATA(NO1)	既存のみ
(3)	SP *.EXAMPLE.DATA(NO2)	新規・既存

SPコマンドの入力方法は、SASプログラムが一時的であるか、又一時的でない場合にも新規か

既存かにより異なる。表4. 1にはそれぞれの場合の入力方法が示されている。表の中で(2)と(3)のデータセット名はSASプログラムが入っているデータセット名である。

以下、簡単なSASプログラムを使って使用例を示す。なお、ユーザが入力すべき部分にはアンダラインを付してある。以後も同様である。

(1) SP *

使用例を図4. 2に示す。

結果を出力する場合には、図にあるようにPROCステートメントの後でRUN;を入力してENTERキーを押す。RUN;を入力しないと次のPROCステートメントが入力され、ENTERキーを押した時点で前の結果が出力される。

この方法はプログラムを保存しない。従ってプログラムの修正もENTERキーを押した後では不可能である。エラーメッセージが出力された場合には、その部分を実行しないので再度入力すればよい。

プログラムの実行を終了するためには/*を入力してENTERキーを押す。その後メッセージが出てREADYモードとなる。

```
READY
SP *
.....
1? DATA;INPUT X Y Z;CARDS;
2? 10 20 30
   データ
20? 100 200 300
21? PROC PRINT;RUN;
   結果を端末に出力する
22? PROC MEANS;RUN;
   結果を端末に出力する
23? /*
.....
READY
```

図4. 2 SP * による実行例

(2) SP EXAMPLE. DATA(NO1)

使用例を図4. 3に示す。但し、EXAMPLE. DATA (NO1)に入っているSASプログラムは示されていない。

この方法は既存のデータセットEXAMPLE. DATA (NO1)に入っているSASプログラムを実行し、結果を端末に出力するものである。SASプログラムをデータセットに保存する方法はFORTRANなどのプログラムを保存する方法と同じである(詳細は、手引き〔7〕を参照の事)。

プログラムの修正は、FSサブコマンドで行なうことができる。

(3) SP *.EXAMPLE. DATA(NO2)

SASプログラムを新規のデータセットEXAMPLE.DATA (NO2) に入力し、実行結果を端末に出力する例が図4. 4 に示されている。

データセットは新規でも既存でもよい。SPコマンド入力後、新規の場合は図のようにINPUTモードとなり、既存の場合はEDITモードとなる。新規の場合にはプログラム入力後、EDITモードになる。いずれの場合も、EDITモードでSASと入力すれば結果が端末に出力される。その後、TRY OR ENDのメッセージが出てTRYと入力するとEDITモードとなるから、FSサブコマンドなどで修正が可能となる。

実行を終了させるにはENDを入力すればよい。これによりデータセットが自動的に保存されてREADYモードとなる。

4. 2. 2 データを外部OSデータセットから入力する場合

この場合もSPコマンドを使って同様に実行できるが、違いはデータが登録されているデータセットの名前とファイル名を、SPコマンド入力時に指定することである。以下では、説明の便宜上、入力データセットはすべて区分データセットでデータセットタイプをDATAとしてあるが、もちろん順データセットを使用してもよい。

以下で説明するSPコマンドの入力方法をSASプログラムの形式別に表4. 2に示す。

```
READY
SP EXAMPLE.DATA(NO1)
結果を端末に出力する
READY
```

図4. 3 SP EXAMPLE.DATA (NO1) による実行例

```
READY
SP *.EXAMPLE.DATA(NO2)
.....ASSUMED TO BE NEW
INPUT
00010 DATA;INPUT X Y Z;CARDS;
.....
データ
.....
00020 PROC PRINT;PROC MEANS;
00021 (ENTERキー)
EDIT
SAS
SAVED IN DATA SET 'EXAMPLE.DATA(NO2)
結果を端末に出力する
ENTER TRY OR END
TRY
E
プログラムの修正
SAS
結果を端末に出力する
ENTER TRY OR END
END
.....
READY
```

図4. 4 SP *.EXAMPLE.DATA (NO2) による修正・実行例

表4. 2 結果を端末出力するためのSPコマンド
(データを外部OSデータセットから入力する場合)

	入 力 方 法	SASプログラム
(1)	SP * F(FF) DA(AB.DATA(P1))	一時的
(2)	SP * F(YES)	一時的
(3)	SP CD.DATA(N1) F(FF) DA(AB.DATA(P2))	既存のみ
(4)	SP CD.DATA(N2) F(YES)	既存のみ
(5)	SP *.CD.DATA(N3) F(FF) DA(AB.DATA(P3))	新規・既存
(6)	SP *.CD.DATA(N4) F(YES)	新規・既存

表4. 2には6つのSPコマンド入力方法が示されているが、(1)と(2)、(3)と(4)、(5)と(6)はそれぞれ対になって代替的に使われる。各々の対の中でどちらを使うかはデータを入力するデータセットが複数個かどうかによる。データセットが1個の場合にはどちらを使ってもよいが、F、DAパラメータ両方を指定する方法の方が便利である。Fでは入力データセットのファイル名を、DAではデータセット名を指定する。複数個のデータセットを入力する場合には、必ずF(YES)とする。

以下、表4. 2のうち(1)、(4)、(5)の入力方法について説明する。

(1) SP * F(FF) DA(AB.DATA(P1))

データをデータセットAB.DATA(P1)から入力し、一時的SASプログラムを実行する。図4. 5に使用例を示す。繰り返し述べたように、Fパラメータのファイル名とINFILEステートメントのファイル名が同一であることに注意されたい。他は、図4. 2の説明と同じなので省略する。

(4) SP CD.DATA(N2) F(YES)

外部OSデータセットからデータを入力し、既存のSASプログラムを実行する。使用例が図4. 6に示されている。ここではデータが2つのデータセットAB.DATA(Q1)、AB.DATA(Q2)から入力されており、それぞれのファイル名をF1、F2としてある。

このようなSPコマンドの入力では、図のようにコマンド入力後、ファイル名入力、及びデータセ

ット名入力の指令が交互に出てくる。すべてを入力後、ENDとすれば実行結果が端末に出力される。詳細については図4. 3の説明を参照されたい。

(5) SP *.CD.DATA(N3) F(F
F) DA(AB.DATA(P3))

SASプログラムを新規のデータセットCD.DATA(N3)に入力し、データセットAB.DATA(P3)から入力されたデータを使って実行する例が図4. 7に示されている。詳細については図4. 4の説明と同じなので省略する。

4. 3 TSSによる入力と実行（プリンタ、NLP出力の場合）

本節では、SPコマンドを使ったTSSによる実行結果をプリンタ、あるいはNLPへ出力する方法について説明する。

4. 3. 1 プリンタに出力する場合

SPコマンドでPRパラメータを指定し、

() 内にユーザ指定名（英字で始まり、残りは英数字の8文字以内）を入力する。例えば

SP CD.DATA(N1) F(FF) DA(AB.DATA(P2)) PR(AA)

とする。これにより、データがAB.DATA(P2)から入力され、既存のデータセットCD.DATA(N1)に入っているSASプログラムが実行され、結果がプリンタへ出力される。

なお、出力先は自動的に最寄りのプリンタとなる。上記以外の形式のSPコマンドでも、PRパラメータを指定すれば結果はすべて最寄りのプリンタに出力される。

```
READY
SP * F(FF) DA(AB.DATA(P1))
.....
1? DATA; INFILE FF; INPUT X Y Z;
2? PROC PRINT; RUN;
   結果を端末に出力する
3? PROC MEANS; RUN;
   結果を端末に出力する
4? /*
```

図4. 5 SP * F(FF) DA(AB
.DATA(P1))による実行例

```
READY
SP CD.DATA(N2) F(YES)
ENTER FILE(DD) NAME OR END
F1
ENTER DATA SET NAME
AB.DATA(Q1)
ENTER FILE(DD) NAME OR END
F2
ENTER DATA SET NAME
AB.DATA(Q2)
ENTER FILE(DD) NAME OR END
END
   結果を端末に出力する
READY
```

図4. 6 SP CD.DATA(N2)
F(YES)による実行例

4. 3. 2 NLPに出力する場合
まず出力結果を保存するデータセッ
トをSPコマンドのOUTパラメータ
で確保する。例えば

```
SP CD.DATA(N1) F(FF)
DA(AB.DATA(P2)) OUT(B
B)
```

とする。ここでOUTパラメータの
()内は、PRパラメータの場合と
同様にユーザ指定名である。この後、
次節で説明するBコマンドのPSTOL
LP機能を使って

```
READY
SP *.CD.DATA(N3) F(FF) DA(AB.DATA(P3))
.....ASSUMED TO BE NEW
INPUT
00010 DATA; INFILE FF; INPUT X Y Z;
00020 PROC PRINT; PROC MEANS;
00030 (ENTERキー)
EDIT
SAS
    結果を端末に出力する
    ENTER TRY OR END
TRY
E
    プログラムの修正
SAS
    結果を端末に出力する
    ENTER TRY OR END
END
.....
READY
```

図4. 7

```
B PSTOLP D(BB)
```

```
SP *.CD.DATA(N3) F(FF)
DA(AB.DATA(P3))
```

による実行例

とする。実行結果は上記のSPコマン
ドによりBB.OUTLISTに出力されているので、BコマンドのDパラメータでもBBを指定す
る。なお、Bコマンドの詳細については、センター発行の手引き「TSSユーザのための便利なコマ
ンド(その1)」 車古正樹著を参照されたい。

NLPへの出力については、次節で別の方法を説明する。

4. 4 Bコマンドによるバッチ処理

本節では、BコマンドのSASJOB機能によりSASプログラムをバッチ処理する方法について
述べる。結果の出力はNLP、LP、端末、プリンタのいずれにも行なうことができる。各々の出力
方法の詳細については手引き〔7〕を参照されたい。

4. 4. 1 一時的SASプログラムの場合

データを外部OSデータセットから入力して一時的SASプログラムを実行する例を以下に示す。

```
READY
B SASJOB D(/YES)
```

```

: ENTER PROGRAM STATEMENT OR END
:  DATA;INFILE FF;INPUT X Y Z;
:  PROC PRINT;PROC MEANS;
:  END
: ENTER FILE(DD) NAME OR END
:  FF
: ENTER DATA SET NAME
:  AB.DATA(Q1)
: ENTER FILE(DD) NAME OR END
:  END
READY

```

上の例ではデータをAB . DATA (Q1) から入力している。データをSASプログラムと共に入力する場合は、単に

B SASJOB

を入力して、以後上の例のようにSASプログラムを入力して行けばよい。

4. 4. 2 既存のSASプログラムの場合

SASプログラムが一時的でない場合にはBコマンドは既存のものに限り、マッチ処理を行なう。例えば、データをAB . DATA (R1) から入力して、既存のデータセットEXAMPLE . DATA (P1) に入っているSASプログラムを実行するには次のようにする。

```

READY
B SASJOB D(EXAMPLE . DATA(P1)/YES)
: ENTER FILE(DD) NAME OR END
:  FF
: ENTER DATA SET NAME
:  AB . DATA(R1)
: ENTER FILE(DD) NAME OR END
:  END
READY

```

以上、BコマンドのSASJOB機能を使った実行方法について述べたがBコマンドでは自動的に次の指定を行なう。

- (1) ジョブ識別子をSとする。
- (2) ジョブクラスをBとする。
- (3) ジョブに関する情報やSASプログラム実行に関する情報を出力する。

以上の点を変更したい時は、Dパラメータで次のようにすればよい。

B SASJOB D(JOB#ACY/EXAMPLE.DATA(P1)/YES)

ここで、JOB#ACYのうち最後の3文字が上述の(1)～(3)を変更するためのもので、これによりジョブ識別子がA、ジョブクラスはCとなる。又、最後の文字Yはメッセージクラスを表わす。Yを指定するとジョブ情報などを出力せず結果だけを出力する。Yの代わりに7を指定すればすべての情報も出力されるが、その場合にはSORPコマンドで出力するものを選択するのが望ましい。

さらに、リージョンサイズをRパラメータで指定できる。例えば

B SASJOB D(JOB#AC7/EXAMPLE.DATA(P1)/YES) R(1024K)

とすればよい。

第5章 SASデータセットの登録とSASデータベース

大量のデータを何度も処理、分析する場合は、データを外部SASデータセットとして登録する方がよい。外部データセットとして登録すれば、そのたびにDATAステップを実行する必要もなくなり、システムに対する負荷も減少するしユーザの手間も省ける。

本章では、SASデータセットを登録する方法、及びこれに関連したSASプロシージャについて説明する。

5.1 SASデータベース

SASデータセットはSASデータベースに登録される。SASデータベースはSASライブラリとも呼ばれるが、SASデータセットだけをいくつでも含むことができる。SASデータベースの構成を図5.1に示す。

図5.1からわかるように、SASデータベースのファイル構成は区分データセットとよく似ており、各々のデータセットはメンバーとみなすこともできるが厳密には異なる。従って、SASデータセットのアクセス、消去、複写、圧縮などはそのためのSASプロシージャを使わなければならない。例えばTSSのDELETEコマンドで特定のSASデータセットを消去しようとするれば、登録されている他のSASデータセットもすべて消去されてしまう。

SASデータベースには英字で始まり、残りは英数字の名前が付けられる。間にピリオドを付ければ、ピリオドを含めて37文字以内のものでよい。但し、ピリオド間は8文字以内である。例えば、KANAZAWA.SASDATA.ANALYSISのように付ける。

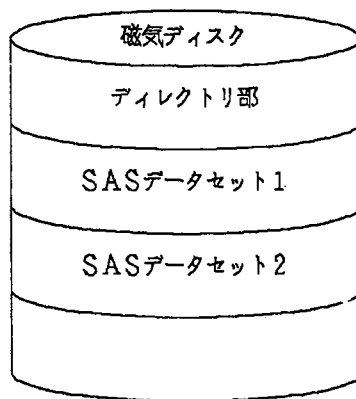


図5.1 SASデータベースの構成

5.2 SASデータセットの登録

本節では、バッチあるいはTSSによりSASデータセットを登録する方法について述べる。

今、20個の変数X1～X20に関するデータが外部OSデータセット EXAMPLE.DAT A (NO1) に入っているものとして、これをSASデータベースに登録するものとして。

5. 2. 1 バッチによる登録

新規のSASデータベースKANAZAWA.UNIに、データセットEXAMPLE.DAT A (NO1) に入っているデータをSASデータセットSURVEYとして登録するためのジョブストリームを図5. 2に示す。

```
//AB9999A JOB ,PASS=XXXXXXXX,CLASS=X
// EXEC SAS
//FF DD DSN=AB9999.EXAMPLE.DAT A(NO1),DISP=SHR           #1
//GG DD DSN=AB9999.KANAZAWA.UNI,DISP=(NEW,CATLG),         #2
//      SPACE=(TRK,(10,10)),UNIT=SYSDA                    #3
//SAS.SYSIN DD *
DATA GG.SURVEY;                                           #4
  INFILE FF;                                              #5
  INPUT  X1-X10/X11-X20;                                  #6
/*
//
```

図5. 2

バッチによるSASデータセット登録のためのジョブストリーム

☆図5. 2の説明

#1：入力データセットを指定するためのDD文である。第4章1節の#3と同様である。

#2、3：SASデータベースを確保するためのDD文である。DISP、SPACE、UNITパラメータを指定する。これらについてはセンター発行の手引き「FORTRANユーザのためのジョブ制御文とデータセット」 山下邦弘著を参照されたい。

#4：DATAステートメントであり、2語のデータセット名を指定する。GGはSASデータベースのファイル名、SURVEYはSASデータセット名である。

#5：外部OSデータセットからの入力をSASに知らせるINFILEステートメントで、FFは入力データセットのファイル名である。

#6：INPUTステートメントの入力。ここではオブザベーションが2行にわたって入力されている。

SASデータベースKANAZAWA.UNIが既存の場合には#2、3を次のように変更すればよい。

//GG DD DSN=AB9999.KANAZAWA.UNI,DISP=OLD

ここでDISPパラメータはOLDを指定することに注意されたい。

SASデータセットの登録は、図5. 2のようにDATAステップで行なわれるだけでなく、プロシージャによってはPROCステップ内でも可能である。その例については次の小節で述べる。

5. 2. 2 SPコマンドによる登録

図5. 2に示されたバッチ処理と同じジョブをSPコマンドで行なった場合のジョブストリームを図5. 3に示す。

☆図5. 3の説明

#1 : SPコマンドの入力。外部データセットを使って一時的SASプログラムを実行する。

#2 : #3と対になっており、入力データセットのファイル名を指定する。

#3 : 入力データセット名を指定する。

#4 : #5と対になっており、SASデータベースのファイル名を指定する。

#5 : SASデータセットを登録するSASデータベース名を指定する。

#6 : すべてのファイル名、データセット名を入力したのでENDと入力する。

#7 : DATAステートメントの入力。データセット名は2語であり、GGがファイル名、SURVEYがSASデータセット名である。GGは#4にあるSASデータベースのファイル名と同じでなければならない。

#8 : INFILEステートメントの入力。ファイル名FFは#2にある入力データセットのファイル名と同じでなければならない。

```
READY
SP * F(YES)                                #1
.....
ENTER FILE(DD) NAME OR END
FF                                           #2
ENTER DATA SET NAME
EXAMPLE, DATA(NO1)                         #3
ENTER FILE(DD) NAME OR END
GG                                           #4
ENTER DATA SET NAME
KANAZAWA, UNI                              #5
ENTER FILE(DD) NAME OR END
END                                           #6
.....
1? DATA GG, SURVEY;                        #7
2? INFILE FF;                               #8
3? INPUT X1-X10/X11-20;
4? /*
.....
READY
```

図5. 3 TSSによるSASデータセット登録のためのジョブストリーム

図5. 3では一時的SASプログラムによりSASデータセットを登録したが、恒久SASプログラムを実行して登録する場合はSPコマンドでデータセット名を指定すればよい。例えば、SASプログラムが新規、あるいは既存の時は、そのデータセット名をABC.DATA(A1)として

SP *.ABC.DATA(A1) F(YES)

とすればよい。ABC.DATA(A1)が既存の時は

SP ABC.DATA(A1)

としてもよい。SPコマンド入力後のジョブストリームは第4章で示したので省略する。

図5. 2及び図5. 3ではDATAステップで作成したSASデータセットを登録する例を示したが、プロシージャによってはPROCステップ内でも可能である。例えばオブザベーションを再配列するためのSORTプロシージャを考えよう。今、入力データセットEXAMPLE.DATA (NO1) の変数X1が取る値の大小順に、小さい方からオブザベーションを配列したものをSASデータセットとして登録するものとしよう。そのためには、図5. 3の#7以降のSASプログラムを次のように入力すればよい。

```
1? DATA;INFILE FF;  
2? INPUT X1-X10/X11-X20;  
3? PROC SORT OUT=GG.SURVEY;  
4? BY X1;RUN;  
5? /*
```

SORTプロシージャ以外にも実行結果をSASデータセットとして登録できるプロシージャがいくつかあるが、これについては第6章以降で説明する。

5. 2. 3 Bコマンドによる登録

Bコマンドによる登録は、SASデータベースが新規でも既存でも同一の方法で行なうことができる。新規の場合は19トラックが自動的に確保される。トラック数を指定する場合、例えば新規のSASデータベースKANAZAWA.UNIを10トラックにする場合はSPコマンドの中でT1パラメータを使って

```
SP FILE DA(KANAZAWA.UNI) T1(10)
```

とする。

以下、SASデータベースKANAZAWA.UNIに対して、図5. 2、図5. 3と同じジョブをBコマンドで行なった場合のジョブストリームを図5. 4に示す。

図5. 4の内容は図5. 2、あるいは図5. 3の説明を読めば理解できると思われるので、その説明は省略する。

既存の恒久SASプログラムにより登録する場合は、Bコマンドのパラメータでそのデータセット名を指定すればよい。例えば

```
B SASJOB D(ABC.DATA(A1)/
YES)
```

とする。これ以降のジョブストリームは第4章で示したので省略する。

5. 3 外部SASデータセットの処理、分析

前節の方法でSASデータセットを登録すれば、処理、分析のためのSASプログラムは非常に簡単になる。

以下、SASデータセットSURVEYがSASデータベースKANAZAWA.UNIに登録されているとして、バッチ及びTSSによる実行方法を説明する。
なお、実行結果の出力は第4章で述べた方法と同じなので省略する。

5. 3. 1 バッチによる実行

図5. 5にバッチによる実行例のジョブストリームを示す。ここではMEANSプロシージャを実行している。

外部SASデータセットを何ら変更せず処理、分析する場合は、図5. 5のようにSASプログラムはPROCステップだけからなってもよい。データセットの一部を変更して実行する場合は、まず図5. 5の中のSASプログラムを

```
DATA;SET FF.SURVEY;
```

として、以後第2章7節で説明したプログラミング・ステートメントなどを使って変更する。そしてこのDATAステップで作成されたSASデータセットに対してプロシージャを実行すればよい。

複数の入力データセットがある場合には、DD文を追加すればよい。

```
READY
B SASJOB D(/YES)
:ENTER PROGRAM STATEMENT OR END
: DATA GG,SURVEY;INFILE FF;
: INPUT X1-X10/X11-X20;
: END
:ENTER FILE(DD) NAME OR END
: FF
:ENTER DATA SET NAME
: EXAMPLE,DATA(NO1)
:ENTER FILE(DD) NAME OR END
: GG
:ENTER DATA SET NAME
: KANAZAWA.UNI
:ENTER FILE(DD) NAME OR END
: END
.....
READY
```

図5. 4 BコマンドによるSASデータセット登録のためのジョブストリーム

```
//AB9999A JOB ,PASS=XXXXXXXX,CLASS=X
// EXEC SAS
//FF DD DSN=AB9999.KANAZAWA.UNI,DISP=SHR
//SAS,SYSIN DD *
PROC MEANS DATA=FF, SURVEY;
/*
//
```

図5. 5 バッチによる実行例
(外部SASデータセット入力の場合)

5. 3. 2 SPコマンドによる実行

図5. 5と同じジョブをSPコマンドで実行する例を図5. 6に示す。

複数の入力ファイルがある場合や、恒久SASプログラムの実行については第4章で述べた方法でできるので省略する。端末以外への出力についても同様である。

5. 3. 3 Bコマンドによる実行

図5. 5及び図5. 6と同じジョブをBコマンドで実行した場合を図5. 7に示す。

恒久SASプログラムの実行も同様なので省略する。

5. 4 SASデータセット管理のためのプロシージャ

本節ではSASデータセットを管理するためのいくつかのプロシージャについて説明する。これらのプロシージャはSASデータセットの削除、改名、コピーなどに関連したものである。本章1節でも述べたようにTSS用のDELETEコマンドなどを用いるとデータベースが破壊されるので注意されたい。以下で説明するプロシージャの中には一時的

SASデータセットに対しても実行できるものもあるが、主として外部SASデータセットに対して有用なものである。従って、各々のプロシージャの説明でDATA=データセット名で指定されるデータセット名は通常2語の名前となる。

☆CONTENTSプロシージャ

- ・SASデータセットの物理的特性や内容をプリントするためのプロシージャである。
- ・一般形は

PROC CONTENTS DATA=データセット名 オプションの並び;

```
READY
SP * F(FF) DA(KANAZAWA.UNI)
.....
1? PROC MEANS DATA=FF, SURVEY; RUN;
    結果を端末に出力する
2? /*
.....
READY
```

図5. 6 SPコマンドによる実行例
(外部SASデータセット入力の場合)

```
READY
B SASJOB D(/YES)
: ENTER PROGRAM STATEMENT OR END
: PROC MEANS DATA=FF, SURVEY;
: END
: ENTER FILE(DD) NAME OR END
  FF
: ENTER DATA SET NAME
  KANAZAWA, UNI
: ENTER FILE(DD) NAME OR END
  END
.....
READY
```

図5. 7 Bコマンドによる実行例
(外部SASデータセット入力の場合)

追加ステートメント;

である。

- ・オプションには以下のものがある。

DIRECTORY: SASデータセットが登録されているSASデータベースのディレクトリ部の情報を出力する。

HISTORY: SASデータセットの履歴情報を出力する。

SHORT: SASデータセットの変数だけをリストする。

- ・追加ステートメントには標題を付けるためのTITLEステートメントがある。
- ・以下にTSSによる使用例を示す。

```
SP * F(FF) DA(KANAZAWA.UNI)
```

.....

```
1? PROC CONTENTS DATA=FF.SURVEY;RUN;
```

結果を端末に出力する。

```
2? /*
```

- ・上記の例で、SASデータベースKANAZAWA.UNIに含まれるすべてのSASデータセットの内容を記述するためには

PROC CONTENTS DATA=FF.__ALL__;とすればよい。

☆COPYプロシージャ

- ・外部SASデータセットを別のSASデータベースにコピーするためのプロシージャであり、空き領域を取り除くために有用である。

- ・一般形は

PROC COPY IN=入力ファイル名 OUT=出力ファイル名;追加ステートメント;

である。入力ファイル名はコピーされるSASデータセットが入っているSASデータベースのファイル名であり、出力ファイル名は新しく登録するデータベースのファイル名である。

- ・上記の入力形式によりすべてのSASデータセットがコピーされる。
- ・特定のデータセットだけをコピーするためには次のような追加ステートメントを使う。

SELECT SASデータセット名の並び;指定されたデータセットだけをコピーする。

EXCLUDE SASデータセット名の並び;指定されたデータセット以外のすべてをコピーする。

☆DATASETSプロシージャ

- ・外部SASデータセットのリスト、削除、改名などのためのプロシージャである。

- ・一般形は

PROC DATASETS DDNAME=ファイル名 オプションの並び;

追加ステートメント;

である。ファイル名には、SASデータベースのファイル名を指定する。

- ・オプションには次のものがある。

KILL: SASデータベース内のすべてのデータセットを削除する。

NOLIST: SASデータセットの内容を出力しない。

- ・追加ステートメントは次の通りである。

DELETE SASデータセット名の並び; 指定されたデータセットを削除する。

KEEP SASデータセット名の並び; 指定されたデータセット以外を削除する。

CHANGE 古い名前=新しい名前; SASデータセット名を変更する。

☆DELETEプロシージャ

- ・SASデータセットを削除するためのプロシージャである。

- ・一般形は

PROC DELETE DATA=データセット名の並び;

である。データセットは一時的なものでもよい。

- ・単にPROC DELETE; とすれば直前に作成されたSASデータセットが削除される。

- ・SASデータベース内のすべてのデータセットを削除するには

PROC DELETE DATA=ファイル名._ALL_とすればよい。

☆EDITORプロシージャ

- ・SASデータセットの編集のためのプロシージャである。

- ・一般形は

PROC EDITOR DATA=データセット名;

である。データセットは一時的なものでもよい。

- ・TSSによりプロシージャを実行するためのジョブストリームを以下に示す。

READY
SP * F(FF) DA(KANAZAWA.UNI)

.....

1? PROC EDITOR DATA=FF.SURVEY;RUN;
ENTER COMMANDS OR 'HELP'

2? LIST 10,20 X1,X2;

10番目から20番目のオブザベーションのうち、X1、X2だけをリストする。
変数名を省略すればすべてのオブザベーションがリストされる。

3? FIND 10, LAST X1=1000;

10番目から最後のオブザベーションの中でX1の値が1000のものを探して、
最初のオブザベーション番号を出力する。

4? REPLACE 15 X1=2000;

15番目のオブザベーションでX1の値を2000にする。

5? ADD X1=10 X2=20 X3=30;

データセットの最後にオブザベーションを追加する。指定されない変数は欠値となる。

6? END

EDITORプロシージャから出て、他のステートメントを入力できる。

・上で示したLIST、FINDなどはコマンドと呼ばれる。これら以外の主要なコマンドは次の通りである。

UP m;現在のオブザベーション位置からmだけ上へ移る。

DOWN n;現在のオブザベーション位置からnだけ下へ移る。

DELETE m,n;mからn番目のオブザベーションに欠値を割り当てる。

VERIFY ON;REPLACEコマンドにより変更された結果を出力する。

☆RELEASEプロシージャ

・SASデータベースの最後の空き領域を解放する。

・一般形は

PROC RELEASE DDNAME=ファイル名 オプション;

である。

・オプションには次のものがある。

TRACKS=トラック数：SASデータベースが含むべきトラック数を指定する。指定された値がSASデータベースのサイズより小さい場合は無視される。